MOBILE AGENT TECHNOLOGY USING AGLETS AS AN ALTERNATIVE FOR INTELLIGENT DISTRIBUTED PROGRAMMING

Ng Melissa Angga

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Surabaya,
Jalan Raya Kalirungkut, Surabaya
melissa@ubaya.ac.id

ABSTRACT

Agent based technology was brought at the first place to assist computer user, especially the novice one, doing their task which were becoming more complex due to the growth of computer technology, including internetworking. One potential development in software agent technology is the mobile software agent technology which accommodate a small program acting as an agent and leap from one place to another based on its own need. AGLETS, which created by Danny Lange and Mitsuru Oshima can be used as an alternative to develop the mobile agents.

Keywords: *agent, mobile agent, AGLETS*

INTRODUCTION

Distributed system was introduced to share the resource in order to reduce the server workload by spread the process needs to be done [1]. However, with the limitation of the network connection while the work across network grew up rapidly, classic distributed system facing some problems especially to handle the network traffic. And that is when the mobile agent technology came to solve the problem faced by previous distributed system. Using mobile agent technology, a capsulation of information and small program could be dispatch to another host and run locally there [7]. This approach would in return reduce the traffic over the network. Moreover, since the computing process has been brought to the target machine, it would still works perfectly even though the source machine has been shut down.

One example of the problem that could be addressed using the mobile agent technology is the travel inquiry process. To create a list of travel itinerary, one commonly needs to collect the information about the flights, hotels, and other information from many servers which each represent one hotel or one flight agent. This paper would take a look on the opportunity laid inside the mobile agent technology and try to adapt it using AGLETS platform on the problem of travel inquiry process.

The Origin of Agent Technology

With the fast growth of computer technology, especially after the internet era where people connected globally, the complexity of the problem and task need to do by human hand in hand with the computer machine raised as well [8]. In this state, the agent based technology, which is a technology to assist people doing their task using the computer, take place, as a useful partner, particularly for the novice computer user going through their task. Agent technology is one step ahead the object technology. Even though it was said that the agent is merely an active [3] and independent object (to distinct it with the object created by object oriented technology), actually there is still no exact definition of an agent [4]. However, the agent paradigm itself recognized through its characteristics, as explained on the following passage.

Shiao stated that the agent is a product of artificial intelligent technology combining with the network technology [9]. More over, an agent is an entity that used its artificial intelligence capability to react behalf of another entity autonomously, that is, under its own control (Figure 1).

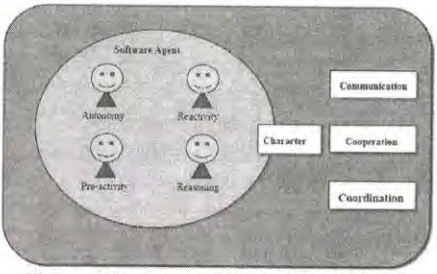

Figure 1The Essence of a Software Agent

The reaction of an agent sometimes is reactive as a respond to its environment, but sometimes is proactive as well, that is, the initiative came from the agent itself. The proactive characteristic came as a result since the agent's behavior designated to reach the goal [11]. In shorter definition, Lange simply concludes an agent as a program that helps and performs behalf of the user [7]. The agent must be situated in a specific execution environment, that is, if the environment

ISBN 978-979-

changed the agent could be no longer exist [2], and the agent utilized itself by accepting the task from the user [7].

Mobile Agent

Distributed system was built with the need of resource sharing in mind, as Colouris [1] stated. The word 'resource' denotes to not only hardware such as printers, moreover it could refer to some software-built product such as databases, files, components and objects that need to be shared across the network. Using distributed system, enterprise could reduce the cost by distributing process through inexpensive systems while reducing server workload.

However, there are some problems with the classic type of distributed systems. One obstacle is the fact that distributed systems usually need to follow and exchange some protocols while establishing the communication between machines in order to finish a task [9]. This multiple interaction between machines using these protocols would inevitably resulting lots of network traffic. Another classic trouble with traditional distributed system is that the system often depending too much on network connection [7]. The assignment could be failed if the connection has failed to accomplish or drop off in the middle way. Later on come a new alternative way in distributed computing which is believed could solve the problems of the data trafficking in old school distributed system. This new wave is the mobile agent technology which is a merger between intelligent system and internetworking.

The idea of the mobile agent concept was brought at the first place in early 1990's when a commercial system called Telescript was publicized by General Magic Inc. This software was presented as a mobile unit which could roam and communicate between agents or resources sited in different hosts in varied network [9]. Thus, one can conclude the definition of the mobile agent as an independent mobile program which able to migrate from one host to another [9] independently at the appropriate time, carrying small information and the computation itself, with the single objective to finish particular task. As an agent, it has the autonomy to decide the next state to process, the target host to travel, or the right time to move or return.

A mobile agent application usually constructed from several mobile agents, where each of them would leap to another host which has appropriate environment and provides information needed to finish the main task [9]. Each agent could interact with other agents in the middle of the computational process. Practically the road to perform the main task consists of several parts. For every part, once the mobile agent finish doing it, the mobile agent would chose the next step to do autonomously. This iteration would be repeated until the task has been done. Known as the merger product of the intelligent system in the shape of software agent computing and the distributed computing [9], mobile agent technology significantly different from the classic network computing. It is not an applet, the program that entirely works on one host after traveling from server to client (one way) based on user request [9], while mobile agent could work on multiple hosts and moved back and forth from server to client and likewise.

Furthermore, mobile agent cannot be seen as an RPC as well, since mobile agent has the autonomy to roam continuously between hosts or servers and choose the target host independently [9]. Contrast with the ordinary migration, mobile agent can freely move anywhere anytime depends on its needs. There are two kinds of agents, one is the stationary agent that resides in a host and do the communication with the other agent on its fixed position [7]. The other kind is the mobile agent which not just stands still on its place while negotiating with other agents. On the contrary, this type of agent is not tied to the place where it first created. It has the freedom to transport from one host to another. The bundle it brings while transporting itself is some information from the last host (usually called the state) and its own computation code. These two kinds of agents at the same environment could shared information and negotiate between each other.

The mobility aspect in mobile agent technology would be an addition to the main characteristics of a general agent, which are autonomy, reactivity, proactivity and reasoning [9]. Hence, the main characteristics of mobile agents could be concluded as follows. First characteristic is the unique identity. That is each mobile agent has a distinctive identity as a representative agent that show what the agent do on behalf of the user. Secondly and the best representative character of a mobile agent is the autonomous mobility. A mobile agent always independently could flow from one machine to another. The last characteristic is the moving continuity. The state of any transporting mobile agent needs to always remain the same before and after the traveling conduct.

As a solution to the problem confronted by the ordinary distributed system, mobile agent concept offers some advantages that would be discuss on the following section. The first advantage feature is that mobile agent could minimize the network traffic [7]. Since the common distributed system usually need to send some message back and forth due to the communication protocols, the network has suffered of lots and lots of data trafficking. While, mobile agents offer a concept in which user can send a package consist of the information needed and the small program (which encapsulate the protocol itself) to create the conversation based on the information attached across network. Once this agent arrives to the destination, it will do the whole communication locally from the destination. In addition, mobile agent also diminish the unused raw data need to be sent across network from one host to another since

ISBN 978-979-17186-0-8

the computation to process those data has arrive to the host itself, thus the data processing could be done locally without any obligation to send the data back and forth. One motto that could be attached to the mobile agent is to "move the computation to the data rather than the data to the computation" [7].

The next advantage is the ability to overpower the significant latency that could happened when some communication need to be done across network [7]. This benefit could be achieved because once the agent that carrying the message from the controller has arrived to the destination host, it would work and execute the message immediately.

While some mobile technologies usually depend too much on the network connection, the economic and technical aspects make it almost impossible to establish the uninterrupted connections itself. In this case, the used of mobile agent would be a good solution. One advantage of the mobile agent concept is that the mobile agent would be work in autonomous and asynchronous way [7]. After being transferred across network while carrying the task required to be done, the mobile agent could carry on the task autonomously and asynchronously. That is, there is no need to hold on to the connection. The mobile agent will have to work independently apart from the controller host. The agent would return to the controller with the result of its task later on when the controller has reconnected again.

The next advantage of mobile agent which needs to be highlighted is that this kind of agent would provide a robust fault tolerance system [7]. Shutting down on one machine will not stop the execution since the agent could be informed and thus, had enough time to leap to another host.

Bearing the characteristics discussed previously in mind, the mobile agent technology would be the right solution to do certain kind of task. Mobile agent could create some replicas from itself and spread them across network to collect data from many host [10]. As a representative of the user, the agent could filter the information available in different host, and do some negotiating with the other agents whether they reside on the host machine or they are another mobile agent from different owner.

AGLETS

The Aglets platform was developed in IBM research laboratory. Aglets stands for Agent and Applets [9], an applets with the agent characteristic. It is simply known as a Java based platform for creating a mobile agent.

The whole environment of the Aglets usually consists of an aglet viewer, an aglet server, and the aglets (or agents) themselves [12]. The aglet viewer is like an applet viewer but with more features that enable user to create, retract, activate, deactivate and dispatch the aglets on the client side. One example of the aglet viewer is Tahiti.

The aglets that been created would be hosts by the aglet server. Every aglet usually lives within the context of a host. The context is the place which has the ability to maintain and manage the active aglets in homogeneous execution environment to shield the system of any malicious aglets, thus it is like the workplace environment of an aglet.

The agent move to several hosts based on the routine stated in the itinerary it carry along when it dispatched, do the computation on those hosts, and after finish the task it will return home to the starter host with the result. While it carry on its task, the agent represent as a proxy. The proxy used to avoid the direct manipulation of aglets public methods, furthermore it provide the location transparency, in which the exact position of the agent would be hidden.

The communication between aglets would be held through sending messages. A massage could be viewed as an object being exchanged between agents. Since the aglets sometimes need to communicate with another agent created from unknown source with unknown structure, the communication only can be perform by exchanging message, since the method invocation would not able to be done.

Case Study: Travel Itinerary System

One case that would be a good example about mobile agents application is the travel itinerary application. In this system user need to create a travel itinerary based on some information, such as the travel destination, the traveling date, the budget provided, etc. The mobile agents would bring those information to some host such as the hotel server, the airline server, etc. and negotiate with the stationary agents located in every host to find the best suited room and flight for the travel itinerary. The implementation of the travel itinerary system would be performed using Aglets. To handle the travel itinerary system, some agents should be created on behalf of every party related to the system. This system related to the user who needs to create the best travel itinerary, the hotels who offer some information about room available, and the airline which has the information about the price and the flight available.

For the user, a travel agent would be created as his representative. The travel agent is a mobile agent who will bring the information about the available budget and the preference of the user to the hotel servers and airline servers. It will collect every information, find the offer that match the user need and budget, offer it back to the user, and when the user agree on one offer it will do the negotiation with the chosen hotel and airline. Every hotel server should create its own agent which is a stationary agent. The hotel agent would provide any information about the hotel rooms and prices. It also has the right to negotiate with the travel agent. This hotel agent could also handle the booking process. Similar with the hotel agent, each airline agent is a representative of an airline company and created as a

ISBN 978-979-17186-0-8

stationary agent. Airline agent could give the information about flight available and the price as well. It could also do the negotiation and the booking process.

After collect the information about user preference, the travel agent would start to dispatch itself. The example code on the creation of travel agent is as follow:

```
public void onCreation(Object init)
{
//assume the URL list has available
//go to the first URL on the list
       dispatch(URLList[0]);
```

Inside the onCreation handling, the mobility listener should be added for a mobile agent to catch the mobility events such as arrival. On the case of the travel agent, once it arrives on a host, it would gain the information needed by exchanging message, then move to another host on the URLList. The example code for adding the mobility listener is:

```
addMobilityListener(
new MobilityAdapter()
{
public void onArrival(MobilityEvent e)
{
  Message msg = new Message
    ("Hotel");
  msg.setArg("type","standard");
  msg.setArg("price",hotelPrice);
  msg.setArg("fromDate",startDate);
  msg.setArg("toDate",returnDate);
  msg.setArg("from", fromCity);
  msg.setArg("destination",toCity);
  msg.setArg("ticketprice",
    ticketPrice);
  msg.setArg ("myProxy",
     remoteContext.getAgletProxy(id));
  Object r= proxy.sendMessage(msg);
   //… processing message…
  nextIdx++;
  dispatch(URLList[nextIdx]);
```

On the hotel side, the hotel agent does not need to be dispatch. Each hotel agent only listening to the message send by travel agent and reply the information requested. All the coding would be placed on the handleMessage handler. First the agent need to extract the message sent by the travel agent, then retrieve the information from the hotel database before reply the information as a message back to the travel agent. The example code for the hotel agent side shown below :

```
public boolean handleMessage
   (Message msg)
{
if (msg.sameKind("Hotel"))
{
  Integer price = (Integer)
          msg.getArg("price");
  String fromDate = (String)
          msg.getArg("fromDate");
  String toDate = (String)
          msg.getArg("toDate");

  //…extract other information
  // from the message…

  // get the travel agent
  // (the message sender) proxy
  Enumeration en =
        getAgletContext()
        .getAgletProxies();
  while (en.hasMoreElements())
  {
        AgletProxy proxy =
          (AgletProxy)
          en.nextElement();
    if(proxy.getAgletClassName()
      .equals("TravelAgent"))
        mProxy = proxy;
  }

//… retrieve the requested info
// from database…
//… save the result as HyattResult

// send the result as a message
Message hotelresult = new Message
("HotelResult");
hotelresult.setArg("result",
  HyattResult);

try
{
  mProxy.sendMessage(hotelresult);
}
catch (Exception a)
{System.out.println("Error sending
   reply");}
return true;
}
```

The similar code as the hotel agent example code previously could be apply to the airline agent as well, since they both are stationary agents and their only task are wait for the request message send by the travel agent, gain the information from their own database and send the result back to the travel agent.

CONCLUSION

Regarding several advantages offered by the mobile agent technology to reduce network traffic and support the fault tolerance system, it would be useful to create the travel itinerary system using the Aglets as the mobile agent platform. From the trial on the application, the founding is that there is no need to hold on the network connection. The agent still works perfectly on their host while the connection has been cut off. When the agent needs to retract, it will wait until the connection reestablish.

ISBN 978-979-17186-0-8

DAFTAR PUSTAKA

Colouris, G.; Dollimore, J.; Kindberg, T., 2001, *Distributed systems concepts and design 3rd ed*, Essex, Addison-Wesley

Franklin, S.; Graesser, A., 1996, *Is It an Agent, or Just a Program*, Memphis, Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Language

Iglesias, et al., 1999, *A survey of Agent-Oriented Methodologies*, Proceedings of the fifth International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages

Jennings, N.R.; Wooldridge, M., 1998, *Pitfalls of agent-oriented development*, Minneapolis/St. Paul, Proceedings of the 2nd Int. Conf. on Autonomous Agents

Karjoth, Gunter; Lange, Danny B.; Oshima, Mitsuru, 1997, *A Security Model for Aglets*, Zurich, IBM Research Division

Kotz, David; Gray, Robert S., 1999, *Mobile Agents and the Future of the Internet*, Hanover, New Hampshire, Department of Computer Science/Thayer School of Engineering Darthmouth College

Lange, Danny B.; Oshima, Mitsuru, 1998, *Programming and Deploying Java Mobile Agents with Aglets*, Massachusetts, Addison-Wesley

Maes, P.; Shneiderman, B., 1997, *Direct manipulation Vs. Interface agent, ACM*

Shiao, D., 2004, *Mobile Agent : New Model of Intelligent Distributed Computing*, China, IBM

Venners, Bill, 1997, *Solve Real Problems with Aglets, a Type of Mobile Agent*, Japan, IBM

Wooldridge, M.; Ciancarini, P., 2001, *Agent-Oriented Software Engineering : the State of the Art*, Limerick, Ireland, First International Workshop, AOSE 2000 on Agent Oriented Software Engineering

Xu, Chong; Tao, Dongbin, no date, *Building Distributed Application with Aglet*, Tokyo, IBM Research Lab.